Efficient Learning of Quadratic Variance Function Directed Acyclic Graphs via Topological Layers

Wei Zhou, Xin He, Wei Zhong & Junhui Wang

JOURNAL OF COMPUTATIONAL AND GRAPHICAL STATISTICS
2022, VOL. 00, NO. 0, 1–11
https://doi.org/10.1080/10618600.2022.2069776

Efficient Learning of Quadratic Variance Function Directed Acyclic Graphs via Topological Layers

Wei Zhou[a,b*], Xin He[c*], Wei Zhong[a], and Junhui Wang[b]

[a]Wang Yanan Institute for Studies in Economics, Department of Statistics and Data Science, School of Economics, Fujian Key Lab of Statistics and MOE Key Lab of Econometrics, Xiamen University, Xiamen, China; [b]School of Data Science, City University of Hong Kong, Kowloon, Hong Kong; [c]School of Statistics and Management, Shanghai University of Finance and Economics, Shanghai, China

ABSTRACT
Directed acyclic graph (DAG) models are widely used to represent casual relationships among random variables in many application domains. This article studies a special class of non-Gaussian DAG models, where the conditional variance of each node given its parents is a quadratic function of its conditional mean. Such a class of non-Gaussian DAG models are fairly flexible and admit many popular distributions as special cases, including Poisson, Binomial, Geometric, Exponential, and Gamma. To facilitate learning, we introduce a novel concept of topological layers, and develop an efficient DAG learning algorithm. It first reconstructs the topological layers in a hierarchical fashion and then recovers the directed edges between nodes in different layers, which requires much less computational cost than most existing algorithms in literature. Its advantage is also demonstrated in a number of simulated examples, as well as its applications to two real-life datasets, including an NBA player statistics data and a cosmetic sales data collected by Alibaba. Supplementary materials for this article are available online.

ARTICLE HISTORY
Received November 2020
Accepted April 2022

KEYWORDS
Causality; Non-Gaussian DAG; Quadratic variance function; Structural equation model (SEM)

1. Introduction

A directed acyclic graph (DAG) model plays a crucial role in causal inference, which is widely used to represent casual relationships among random variables, and has a wide range of applications such as genetics, finance, and machine learning (Sachs et al. 2005; Koller and Friedman 2009; Sanford and Moosa 2012). Yet, learning a DAG model from observed data is challenging both methodologically and computationally, largely due to the identifiability issue and the required acyclicity.

Most of the earlier DAG learning approaches in literature ignore the identifiability issue and mainly focus on recovering the Markov equivalent class (Spirtes, Glymour, and Scheines 2000) of the DAG model. For example, the search-and-score algorithm (Chickering 2003; Nandy, Hauser, and Maathuis 2018; Zheng et al. 2018; Zhu, Ng, and Chen 2020) maximized the regularized likelihood for a DAG model with the best score, and the constrained-based method (Spirtes, Glymour, and Scheines 2000; Tsamardinos, Brown, and Aliferis 2006; Kalisch and Bühlmann 2007) first conducted local conditional independence tests to learn the skeleton of the DAG model, and then determined the edge directions based on acyclicity, v-structures, and other available structures. These methods are able to recover the Markov equivalent class of the DAG model under some assumptions, such as the Markov property and the faithfulness condition. To fully identify DAG models, a number of learning algorithms have been developed under various assumptions on the underlying probability distribution of the DAG model, often represented as a structural equation model (SEM). Particularly, Peters and Bühlmann (2014) established the identifiability of linear Gaussian SEM models with equal error variances, and various algorithms have been developed accordingly (Peters and Bühlmann 2014; Chen, Drton, and Wang 2019; Yuan et al. 2019). Following this line, Shimizu et al. (2006, 2011) and Wang and Drton (2020) established the identifiability of linear non-Gaussian SEM model and developed the corresponding learning algorithms, and Bühlmann, Peters, and Ernest (2014) and Peters et al. (2014) established the identifiability of nonparametric SEM models with additive noise or nonparametric additive noise models.

More recently, Park and Raskutti (2018) and Park and Park (2019) studied a general class of non-Gaussian DAG models, denoted as QVF-DAGs, which require that the conditional variance of each node given its parents is a quadratic function of its conditional mean. This assumption provides a natural criterion for determining the causal ordering of nodes without making additional distributional assumptions, and contains many non-Gaussian distributions as its special cases. An over-dispersion scoring (ODS; Park and Raskutti 2018) algorithm is also developed for learning QVF-DAG models, which first estimates the moral graph of the QVF-DAG model and learns the causal ordering of all nodes sequentially, and then determines the parents of each node through some sparse regression models over the nodes which are causally ahead of it. Yet the computational

CONTACT Junhui Wang j.h.wang@cityu.edu.hk School of Data Science, City University of Hong Kong, Kowloon, Hong Kong.
*These authors have contributed equally to this work.
Supplementary materials for this article are available online. Please go to www.tandfonline.com/r/JCGS.

cost of the ODS algorithm is usually expensive even for learning a medium-size QVF-DAG model.

In this article, we propose a computationally efficient learning algorithm for QVF-DAGs based on a novel concept of topological layers. The idea is very intuitive and any DAG model can be reorganized into an equivalent topological structure with multiple layers, which automatically ensures acyclicity as a node can only have children and offsprings in its lower layers. More importantly, QVF-DAGs can be reconstructed via topological layers based on the proposed ratio-based criterion, which can be used to hierarchically determine the membership of each layer. Once the layers are determined, parents of each node can be recovered by applying some sparse regression methods over all nodes in its upper layers.

Compared with the ODS algorithm in Park and Raskutti (2018), the proposed learning algorithm has a number of advantages. First, the topological layers in the proposed algorithm are unique for any given QVF-DAG model, whereas the ODS algorithm needs to estimate the indeterministic causal ordering of nodes. Second, the computational cost of the proposed algorithm is much less than that of the ODS algorithm, especially in some wide yet shallow QVF-DAG models, such as a hub graph, which has attracted tremendous interest in network analysis as it is a main building block for many network architectures. More precisely, to learn a hub graph with p nodes and n samples as in Figure 2, the computational complexity of our proposed algorithm is of order $O(np)$, which is much more efficient than the ODS algorithm, whose computational complexity is of order $O(np^2)$ (Park and Raskutti 2018).

The rest of this article is organized as follows. Section 2 introduces QVF-DAGs and the concept of topological layers, and the reconstruction criterion of QVF-DAGs based on topological layers. Section 3 provides the details of the proposed learning algorithm for QVF-DAGs. Numerical experiments on simulated examples are conducted in Section 4 to demonstrate the advantages of the proposed algorithm compared with some existing competitors. Section 5 applies the proposed algorithm to analyze two real-life datasets, including an NBA player statistics data and a cosmetic sales data from Alibaba company. A summary is given in Section 6. Some computational details, all the theoretical results, and the corresponding proofs are provided in supplementary materials.

2. QVF-DAG and Topological Layers

2.1. QVF-DAG

A DAG model is widely used to encode joint distribution of a random vector $(X_1, \ldots, X_p)$. Precisely, let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ denote a DAG, where $\mathcal{V} = \{1, \ldots, p\}$ represents a set of nodes associated with variables $X = (X_j)_{j\in\mathcal{V}}$, and $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ denote a set of directed edges without directed cycles. A directed edge from node k to node j is denoted as $k \to j$, and then node k is a parent of node j, and the set of node j's parents is denoted as pa_j. Let $X_{\text{pa}_j} := \{X_k : k \in \text{pa}_j \subset \mathcal{V}\}$ and $X_{\mathcal{S}} := \{X_k : k \in \mathcal{S} \subset \mathcal{V}\}$ for any subset $\mathcal{S}$ of $\mathcal{V}$. We assume that the joint distribution $P(X)$ satisfies the Markov property with respect to $\mathcal{G}$, and thus it allows for the following factorization,

$$P(X) = \prod_{j\in\mathcal{V}} P(X_j|X_{\text{pa}_j}),$$

where $P(X_j|X_{\text{pa}_j})$ denotes the conditional distribution of X_j given its parents X_{pa_j}.

In this article, we focus on the non-Gaussian DAG models with the quadratic variance function (QVF) property (Park and Raskutti 2018). Specifically, we assume that $P(X_j|X_{\text{pa}_j})$ satisfies the QVF property that there exist some constants β_{j1} and β_{j2} such that

$$\text{var}(X_j|X_{\text{pa}_j}) = \beta_{j1}E[X_j|X_{\text{pa}_j}] + \beta_{j2}\big(E[X_j|X_{\text{pa}_j}]\big)^2. \quad (1)$$

In literature, the natural exponential family with the QVF property has been extensively studied (Morris 1982; Brown, Cai, and Zhou 2010), which further assumes that $P(X_j|X_{\text{pa}_j})$ belongs to the exponential family,

$$P(X_j|X_{\text{pa}_j}) = \exp\Big(\theta_j X_j + \sum_{k\in\text{pa}_j} \theta_{jk}X_kX_j - B_j(X_j) - A_j\Big(\theta_j + \sum_{k\in\text{pa}_j}\theta_{jk}X_k\Big)\Big), \quad (2)$$

where $A_j(\cdot)$ is a log-partition function, $B_j(\cdot)$ is determined by a given distribution in the exponential family, and the parameter $\theta_{jk} \in \mathcal{R}$ represents the effect from node k to node j. As pointed out in Park and Raskutti (2018), the QVF-DAG model is identifiable and both assumptions (1) and (2) are satisfied by many popular non-Gaussian distributions, including Poisson, Binomial, Geometric, Exponential, Gamma, and so on.

2.2. Topological Layers

We introduce a novel concept of topological layers, to reformulate a QVF-DAG model into an equivalent topological structure with multiple layers. Particularly, all root nodes in the QVF-DAG model are assigned to the top layer, and other nodes are assigned to different layers according to their longest distances to a root node. It is also important to point out that the induced topological layers are unique for the given QVF-DAG model, and the parents of each node must belong to its upper layers, thus, automatically assuring acyclicity.

Suppose there are a total of T layers, where T denotes the longest possible distance from some node in the QVF-DAG model to a root node. Let $\mathcal{A}_0$ denote the set of root nodes and isolated nodes in the top layer, $\mathcal{A}_t$ denote the set of nodes in the $(t+1)$th topological layer, and $\mathcal{S}_t = \cup_{d=0}^{t-1}\mathcal{A}_d$ denote all the nodes in the layers above the $(t+1)$th layer. For any node $k \in \mathcal{A}_t$, its parent nodes are denoted as pa_k, and thus $\text{pa}_k \subseteq \mathcal{S}_t$.

Figure 1 provides a toy QVF-DAG model with three topological layers. Note that node 1 is a root node and node 4 is an isolated node, and thus both nodes belong to $\mathcal{A}_0$. Node 2 belongs to $\mathcal{A}_1$ as its longest distance to the root node is 1, but node 3 belongs to $\mathcal{A}_2$ instead of $\mathcal{A}_1$ since its longest path to the root node is $1 \to 2 \to 3$. Note that the topological layers is closely related with the key ideas of the depth-first search (DFS) and breadth-first search (BFS) algorithms (Cormen et al. 2009).

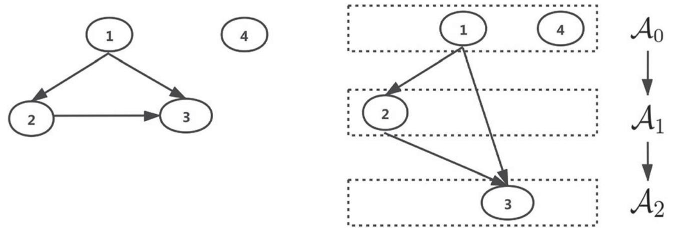

Figure 1. A toy QVF-DAG model in the left panel, and its equivalent topological structure with three layers in the right panel.

Specifically, the number of topological layer for a DAG is defined by the longest distance to a root node, which is similar to the DFS algorithm exploring as far as possible starting from a root node. The idea that nodes with the same distance to a root node are in the same layer is similar to that of the BFS algorithm, which explores all of the neighbor nodes of the current depth before moving to the next depth level.

2.3. Reconstruction of Topological Layers

Let nd_j denote all the non-descendant nodes of node j excluding itself, then the topological layers of a QVF-DAG model can be reconstructed under some mild technical conditions.

Condition 1. For any node $j \in \mathcal{V}$ and any set $\mathcal{S}$ satisfying that $\text{pa}_j \nsubseteq \mathcal{S} \subset \text{nd}_j$, we have $E\big[\omega_j^2(\mathcal{S})\,\text{var}(E[X_j|X_{\text{pa}_j}]|X_{\mathcal{S}})\big] > 0$, and $\omega_j(\mathcal{S}) = (\beta_{j1} + \beta_{j2}E[X_j|X_{\mathcal{S}}])^{-1}$ exists.

Condition 1 is quite general and can be verified for many popular distributions, including Poisson, Binomial, Exponential, Gamma, Geometric, Negative Binomial and so on. The first part of Condition 1 requires all the parents of node j should contribute to its variability, which is more general than that in Park and Raskutti (2018) assuming $\text{var}(E[X_j|X_{\text{pa}_j}]|X_{\mathcal{S}}) > 0$ for all $X_{\mathcal{S}}$, and can reduce to the identifiability condition in Park and Park (2019) when the underlying distribution is indeed Poisson. For illustration, we consider a toy example with $X_1 \sim \text{Poisson}(\lambda)$, $X_2|X_1 \sim \text{Poisson}(\lambda + X_1)$ and $X_3|X_1, X_2 \sim \text{Poisson}(\lambda + X_2 1_{\{X_1 \neq 1\}})$, where $1_{\{\cdot\}}$ is an indicator function. It can be verified that $\text{var}(E[X_3|X_1, X_2]|X_1 = 1) = 0$ and thus the identifiability condition in Park and Raskutti (2018) is violated, but this toy example still satisfies the first part of Condition 1. The second part of Condition 1 requires that β_{j1} and β_{j2} should satisfy $\beta_{j1} + \beta_{j2}E[X_j|X_{\mathcal{S}}] \neq 0$, which ensures that $\omega_j(\mathcal{S}) = (\beta_{j1} + \beta_{j2}E[X_j|X_{\mathcal{S}}])^{-1}$ is well-defined. Moreover, $\beta_{j2} > -1$ is required to rule out some distributions, including Bernoulli and multinomial distributions, which are known to be unidentifiable in literature (Heckerman, Geiger, and Chickering 1995).

Lemma 1. Suppose that $X \in \mathcal{R}^p$ is generated by a QVF-DAG model and Condition 1 is satisfied. For any node $j \in \mathcal{V}$ and any set $\mathcal{S} \subseteq \text{nd}_j$, we have

$$E\big[\text{var}(\omega_j(\mathcal{S})X_j|X_{\mathcal{S}})\big] \geq E[\omega_j(\mathcal{S})X_j],$$

provided that $\beta_{j2} > -1$. The equality holds if and only if $\text{pa}_j \subseteq \mathcal{S}$.

Lemma 1 provides a crucial criterion to reconstruct the topological layers of a QVF-DAG model in a top-down fashion, and its proof is provided in Appendix B. Specifically, if $\text{pa}_j \subseteq \mathcal{S}$, the conditional ratio

$$\mathcal{R}(j, \mathcal{S}) := \frac{E\big[\text{var}(\omega_j(\mathcal{S})X_j|X_{\mathcal{S}})\big]}{E[\omega_j(\mathcal{S})X_j]} \tag{3}$$

should be exactly 1. Motivated by this fact, Theorem 1 shows that the topological layers $\{\mathcal{A}_t\}_{t=0}^{T-1}$ of a QVF-DAG model, defined in Section 2.2, can be exactly reconstructed.

Theorem 1. Suppose that all the conditions of Lemma 1 are satisfied and $\mathcal{A}_0, \ldots, \mathcal{A}_{t-1}$ have been identified with $\mathcal{S}_0 = \emptyset$ and $\mathcal{S}_t = \cup_{d=0}^{t-1}\mathcal{A}_d$. It then holds true that

$$\mathcal{R}(j, \mathcal{S}_t) \begin{cases} = 1, & \text{for any } j \in \mathcal{A}_t; \\ \neq 1, & \text{for any } j \in \mathcal{V}\backslash\{\mathcal{S}_t \cup \mathcal{A}_t\}, \end{cases} \tag{4}$$

for $t = 0, \ldots, T-1$, and thus the topological layers can be exactly reconstructed.

Theorem 1 provides a constructive result for the reconstruction of a general class of non-Gaussian DAG models with the QVF property. Its proof follows from Lemma 1, the criterion in (4), and the assumption that a node's parents should all contribute to its variability. Particularly, for any root or isolated node $j \in \mathcal{V}$ with $\text{pa}_j = \emptyset$, Theorem 1 shows that the topological layer $\mathcal{A}_0$ can be exactly reconstructed by the fact that

$$\mathcal{R}(j, \emptyset) = \frac{E\big[\text{var}(\omega_j(\emptyset)X_j)\big]}{E[\omega_j(\emptyset)X_j]} = \frac{\text{var}(X_j)}{(\beta_{j1} + \beta_{j2}E[X_j])E[X_j]} = 1,$$

for any root or isolated node $j \in \mathcal{V}$, and $\mathcal{R}(j, \emptyset) \neq 1$ otherwise. Further, if the longest distance of a node j to a root node is $t \geq 1$, then $j \in \mathcal{A}_t$ by definition, and Theorem 1 guarantees that $\mathcal{R}(j, \mathcal{S}_t) = 1$ and $\mathcal{R}(l, \mathcal{S}_t) \neq 1$ for any node l contained in lower layers.

3. Proposed Algorithm

In this section, we illustrate the proposed algorithm with natural exponential family assumption in (2), but the algorithm can be adapted to learn a general QVF-DAG model as well. Motivated by Lemma 1 and Theorem 1, learning a QVF-DAG model from the observed data can be decomposed into a two-step procedure, where the topological layers can be reconstructed in a top-down fashion at the first step, and then the directed edges can be recovered by using sparse regression models in a parallel fashion.

3.1. Two-Step Learning Algorithm

Given a training sample $X^n = (X_i^n)_{i=1}^n$ with $X_i^n = (X_{i,1}^n, \ldots, X_{i,p}^n)^T$, we first attempt to estimate the top layer $\mathcal{A}_0$. Specifically, for each node $j \in \mathcal{V}$, we compute the estimated unconditional ratio,

$$\widehat{\mathcal{R}}(j, \emptyset) = \frac{\widehat{\text{var}}(X_j)}{(\beta_{j1} + \beta_{j2}\widehat{E}[X_j])\widehat{E}[X_j]}, \tag{5}$$

where $\widehat{\mathrm{var}}(X_j) = \widehat{E}[X_j^2] - (\widehat{E}[X_j])^2, \widehat{E}[X_j] = \frac{1}{n}\sum_{i=1}^{n} X_{i,j}^n$ and $\widehat{E}[X_j^2] = \frac{1}{n}\sum_{i=1}^{n}(X_{i,j}^n)^2$. By Theorem 1, $\mathcal{A}_0$ can be estimated as $\widehat{\mathcal{A}}_0 = \{j, |\widehat{\mathcal{R}}(j,\emptyset) - 1| \le \epsilon_0\}$ for some small constant $\epsilon_0 > 0$.

Suppose that the topological layers $\widehat{\mathcal{A}}_0, \dots, \widehat{\mathcal{A}}_{t-1}$ have been estimated and $\widehat{\mathcal{S}}_t = \cup_{d=0}^{t-1}\widehat{\mathcal{A}}_d$, we now proceed to estimate $\mathcal{A}_t$. For each node $j \in \mathcal{V}\backslash\widehat{\mathcal{S}}_t$, we compute the estimated conditional ratio,

$$\widehat{\mathcal{R}}(j,\widehat{\mathcal{S}}_t) = \frac{\widehat{E}\left[\widehat{\mathrm{var}}(\widehat{\omega}_j(\widehat{\mathcal{S}}_t)X_j|X_{\widehat{\mathcal{S}}_t})\right]}{\widehat{E}[\widehat{\omega}_j(\widehat{\mathcal{S}}_t)X_j]}, \tag{6}$$

where $\widehat{E}\left[\widehat{\mathrm{var}}(\widehat{\omega}_j(\widehat{\mathcal{S}}_t)X_j|X_{\widehat{\mathcal{S}}_t})\right] = \widehat{E}\left[\widehat{\omega}_j^2(\widehat{\mathcal{S}}_t)\left(\widehat{E}[X_j^2|X_{\widehat{\mathcal{S}}_t}] - (\widehat{E}[X_j|X_{\widehat{\mathcal{S}}_t}])^2\right)\right]$ with $\widehat{\omega}_j(\widehat{\mathcal{S}}_t) = (\beta_{j1} + \beta_{j2}\widehat{E}[X_j|X_{\widehat{\mathcal{S}}_t}])^{-1}$. By Theorem 1, $\mathcal{A}_t$ can be estimated as $\widehat{\mathcal{A}}_t = \left\{j, |\widehat{\mathcal{R}}(j,\widehat{\mathcal{S}}_t) - 1| \le \epsilon_t\right\}$ for some small positive constant ϵ_t. This procedure is repeated until there are no remaining nodes, and then the topological layers of the DAG model are reconstructed.

Once all the topological layers are reconstructed, the directed edges between nodes can be recovered via standard sparse regression models (Meinshausen and Bühlmann 2006; Yang et al. 2015) in a parallel fashion. Specifically, for each node $j \in \widehat{\mathcal{A}}_t$, we conduct a sparse regression of X_j against $X_{\widehat{\mathcal{S}}_t}$, and any nonzero coefficient leads to a directed edge pointing from the corresponding node in $\widehat{\mathcal{S}}_t$ to the node j. This sparse regression procedure can be done for all nodes simultaneously to expedite the computation.

The proposed two-step learning algorithm for a QVF-DAG via topological layer is summarized in Algorithm 1, denoted as the TLDAG algorithm.

Note that the ratio estimation $\widehat{\mathcal{R}}(j,\widehat{\mathcal{S}}_t)$ in Step 2 varies from one distribution to another, and we provide the computational details for Poisson and Binomial DAGs with generalized linear model (GLM) in Appendix A, supplementary materials. A conditional variance estimation procedure for a general QVF-DAG is also provided in Appendix D, supplementary materials. Furthermore, multiple sparse regressions are fitted in Step 4 to recover the parent–child relations, and the choice of the tuning parameters can be guided by Theorem S2 in Appendix C, supplementary materials or via cross-validation. In all the numerical experiments in Section 4, we employ the 5-fold cross-validation to select the optimal values of tuning parameters for the ℓ_1-penalized regressions by using R package "glmnet," which leads to satisfactory numerical performance.

Algorithm 1 TLDAG algorithm

```
1: Input: sample matrix $X^n \in \mathcal{R}^{n\times p}$, $\widehat{\mathcal{S}} = \emptyset$, and $t = 0$;
2: Until $\widehat{\mathcal{S}} = \mathcal{V}$:
     a. For any $j \in \mathcal{V}\backslash\widehat{\mathcal{S}}$, compute the ratio $\widehat{\mathcal{R}}(j,\widehat{\mathcal{S}}_t) = \frac{\widehat{E}\left[\widehat{\mathrm{var}}(\widehat{\omega}_j(\widehat{\mathcal{S}}_t)X_j|X_{\widehat{\mathcal{S}}_t})\right]}{\widehat{E}[\widehat{\omega}_j(\widehat{\mathcal{S}}_t)X_j]}$;
     b. Define $\widehat{\mathcal{A}}_t = \left\{j, |\widehat{\mathcal{R}}(j,\widehat{\mathcal{S}}_t) - 1| \le \epsilon_t\right\}$ and let $\widehat{\mathcal{S}} = \widehat{\mathcal{S}} \cup \widehat{\mathcal{A}}_t$;
     c. $t \leftarrow t + 1$.
3: Let $\widehat{T} = t$.
4: For any node $j \in \widehat{\mathcal{A}}_t$, fit a sparse regression model of $X_j$
   against $X_{\widehat{\mathcal{S}}_t}$ to obtain the estimated directed edges pointing
   to node $j$, denoted as $\widehat{\mathcal{E}}_t = \{(k,j)|k \in \widehat{\mathcal{S}}_t, j \in \widehat{\mathcal{A}}_t\}$
5: Return: $\{\widehat{\mathcal{A}}_t\}_{t=0}^{\widehat{T}-1}$ and $\{\widehat{\mathcal{E}}_t\}_{t=1}^{\widehat{T}-1}$.
```

More importantly, we have established the consistency of the TLDAG algorithm for a Poisson DAG via GLM estimation in Appendix C, supplementary materials and a general QVF-DAG via the conditional variance estimation procedure in Appendix D, supplementary materials, respectively. The theoretical results indicate that the proposed algorithm can exactly recover the true DAG with high probability, with some regularity conditions and properly chosen ϵ_t. Due to the space limit, all the theoretical results, including the consistency of the TLDAG algorithm and its technical proof, as well as the detailed discussion on the suggested tuning method for ϵ_t in Section 3.3 are provided in Appendices C and D, supplementary materials.

3.2. Computational Complexity

In literature, the ODS algorithm (Park and Raskutti 2018) is developed for learning the QVF-DAG models, and the moments ratio scoring algorithm (MRS; Park and Park 2019) extends the ODS algorithm for a special type of QVF-DAGs, yet the computational complexities of both algorithms are of order $O(np^2)$ and $O(np^3)$, respectively, which are still relatively high and difficult to scale up for large-scale QVF-DAGs.

By contrast, the computational complexity of the TLDAG algorithm can be much less than that of ODS and MRS, especially when dealing with shallow QVF-DAGs with $T \ll p$. For example, to learn a hub graph with $T = 2$ as in Figure 2, TLDAG needs to first identify $\mathcal{A}_0$ and $\mathcal{A}_1$, and then reconstruct the parent–child relationship. More precisely, identifying $\mathcal{A}_0$ requires estimation of the unconditional ratio for p nodes, which amounts to the complexity of order $O(np)$, and identifying $\mathcal{A}_1$ requires estimation of the conditional ratio for the remaining $p - 1$ nodes via GLM with only one predictor in $\mathcal{A}_0$, which also amounts to the complexity of order $O(np)$. For parent-child reconstruction, we just need to fit the GLM model for each node in $\mathcal{A}_1$ against the only predictor in $\mathcal{A}_0$, which amounts to the complexity of order $O(np)$. Therefore, the computational complexity of TLDAG is only of order $O(np)$, which is much more efficient than both ODS and MRS algorithms.

In terms of computational complexity of a random graph with T layers, the TLDAG algorithm involves $O(p(T - 1))$ calculations of ratios in Step 2, which is more efficient than ODS and MRS as they both need to calculate $O(p^2)$ ratios to learn the causal ordering. Furthermore, the directed parent-child structure can be recovered in a parallel fashion, and for each node, ℓ_1-regularized GLM regression is fitted using coordinate descent

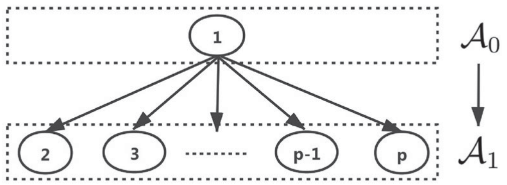

Figure 2. The hub graph for Examples 1 and 2.

(Friedman, Hastie, and Tibshirani 2010; Park and Park 2019), with all the nodes in the upper layers as predictors, leading to the total complexity of order $O\big(\sum_{t=1}^{T-1} n(\sum_{k=0}^{t-1} a_k)a_t\big)$ with a_t denoted as the number of nodes in $\mathcal{A}_t$. It is clear that the computational complexity of TLDAG is the same as that of ODS when $T = p$, and it can be significantly better than ODS when the QVF-DAG has a shallow structure with $T < p$. The detailed numerical comparisons of TLDAG, ODS, and MRS are given in Section 4.3.

3.3. Tuning

The numerical performance of the TLDAG algorithm depends on the layer reconstruction parameter ϵ_t and the tuning parameter in the sparse regression algorithm. Whereas the latter can be determined via cross-validation, we need to modify the stability-based criterion in Sun, Wang, and Fang (2013) to select the optimal parameter ϵ_t for each layer.

The key idea is to measure the reconstruction stability by randomly splitting the training sample into two parts and comparing the disagreement between the two estimated active sets. Specifically, given a value ϵ_t, we randomly split the training sample $\mathcal{Z}^M$ into two parts $\mathcal{Z}_1^M$ and $\mathcal{Z}_2^M$. Then the proposed method is applied to $\mathcal{Z}_1^M$ and $\mathcal{Z}_2^M$ and we obtain two estimated active sets $\widehat{\mathcal{A}}_{1,\epsilon_t}$ and $\widehat{\mathcal{A}}_{2,\epsilon_t}$, respectively. The disagreement between $\widehat{\mathcal{A}}_{1,\epsilon_t}$ and $\widehat{\mathcal{A}}_{2,\epsilon_t}$ is measured by Cohen's kappa coefficient

$$\kappa(\widehat{\mathcal{A}}_{1,\epsilon_t}, \widehat{\mathcal{A}}_{2,\epsilon_t}) = \frac{\Pr(a) - \Pr(e)}{1 - \Pr(e)},$$

where $\Pr(a) = \frac{n_{11}+n_{22}}{p_t}$ and $\Pr(e) = \frac{(n_{11}+n_{12})(n_{11}+n_{21})}{p_t^2} + \frac{(n_{12}+n_{22})(n_{21}+n_{22})}{p_t^2}$ with $n_{11} = |\widehat{\mathcal{A}}_{1,\epsilon_t} \cap \widehat{\mathcal{A}}_{2,\epsilon_t}|$, $n_{12} = |\widehat{\mathcal{A}}_{1,\epsilon_t} \cap \widehat{\mathcal{A}}^c_{2,\epsilon_t}|$, $n_{21} = |\widehat{\mathcal{A}}^c_{1,\epsilon_t} \cap \widehat{\mathcal{A}}_{2,\epsilon_t}|$, $n_{22} = |\widehat{\mathcal{A}}^c_{1,\epsilon_t} \cap \widehat{\mathcal{A}}^c_{2,\epsilon_t}|$, $p_t = p - |\mathcal{S}_t|$, and $|\cdot|$ denotes the set cardinality. The procedure is repeated for B times and the estimated reconstruction stability is measured as

$$\hat{s}(\Psi_{\epsilon_t}) = \frac{1}{B}\sum_{b=1}^{B} \kappa(\widehat{\mathcal{A}}^b_{1,\epsilon_t}, \widehat{\mathcal{A}}^b_{2,\epsilon_t}),$$

where $\widehat{\mathcal{A}}^b_{1,\epsilon_t}$ and $\widehat{\mathcal{A}}^b_{2,\epsilon_t}$ are the estimated active sets in the bth splitting. Finally, we set $\widehat{\epsilon}_t = \min\left\{\epsilon_t : \frac{\hat{s}(\Psi_{\epsilon_t})}{\max_{\epsilon_t} \hat{s}(\Psi_{\epsilon_t})} \geq c\right\}$, where $c \in (0, 1)$ is some given percentage. For illustration, $c = 0.9$ and $B = 5$ are used in all the numerical examples, and yield satisfactory performance. Note that it can be shown that Assumptions 1 and 2 in Sun, Wang, and Fang (2013) can be verified by setting $\lambda = 1/\epsilon_t$, and thus the selection guarantee of the suggested tuning method established in Sun, Wang, and Fang (2013) still holds. More detailed discussion on the stability selection method is provided in Appendix C.1, supplementary materials.

4. Simulated Experiments

In this section, we examine the numerical performance of the TLDAG algorithm, and compare it against some state-of-the-art methods in terms of estimation accuracy of directed edges and computational efficiency. Specifically, five competitors are considered, including the ODS algorithm, the MRS algorithm, a direct linear non-Gaussian DAG method (DLiNGAM; Shimizu et al. 2011), the greedy equivalence search method (GES; Chickering 2003), and the max-min hill climbing method (MMHC; Tsamardinos, Brown, and Aliferis 2006). We implement TLDAG, ODS and MRS in R and the source codes are available in *https://github.com/WeiZHOU23/TLDAG*, implementation of DLiNGAM is available on the author's website *https://github.com/cdt15/lingam*, and GES and MMHC are implemented in the R packages "pcalg" (Kalisch et al. 2012) and "bnlearn" (Scutari 2010), respectively. For GES, the output is a partial DAG, and we follow the treatment in Yuan et al. (2019) and extend the output to DAG for fair comparison. For TLDAG, the tuning parameter ϵ_t's are adaptively chosen for each layer via the stability selection procedure in Section 3.3, where the grid search is conducted over grids $\{10^{-2+0.15s}; s = 0, \ldots, 60\}$.

For comparison metrics, we use Recall, Precision, F1-score and the normalized hamming distance (HM) to evaluate the estimation accuracy. Whereas the first three metrics are standard and popularly used in literature, HM measures the number of edge insertions, deletions or flips needed to transform one graph to another (Tsamardinos, Brown, and Aliferis 2006). Large values of Recall, Precision, and F1-score and small values of HM indicate good estimation accuracy.

In all simulated examples, we generate data for the QVF-DAG model with both hub graphs and random graphs. The conditional distribution of each node given its parents follows either a Poisson distribution $\text{Pois}(\exp(\theta_j + \sum_{k\in \text{pa}_j} \theta_{jk}X_k))$ or a Binomial distribution $\text{Bin}(N_j, \text{logit}^{-1}(\theta_j + \sum_{k\in \text{pa}_j} \theta_{jk}X_k)))$. A nonzero coefficient θ_{jk} indicates a directed edge from node k to node j, and $\theta_{jk} = 0$ otherwise. Therefore, the conditional distribution of each node given its parents indeed belongs to the natural exponential family, and satisfies the required assumptions in (1) and (2).

4.1. Hub Graphs

The hub graph is a special type of DAGs, which consists of a hub node and a number of other nodes, and directed edges only points from the hub node to other nodes. In this section, we consider the following hub graphs, and similar examples have also been considered in Park and Park (2019) and Yuan et al. (2019).

Example 1 (Poisson hub graph). The generated DAG model is depicted in Figure 2. We first generate the hub node X_1 in $\mathcal{A}_0$ from $\text{Pois}(\exp(\theta_1))$, and then X_j in $\mathcal{A}_1$ from $\text{Pois}(\exp(\theta_j + \theta_{j1}X_1))$ for $j = 2, \ldots, p$. The parameters θ_j and θ_{j1} are generated uniformly from $[1, 3]$ and $[0.1, 0.5]$ for $j = 1, \ldots, p$, respectively.

Example 2 (Mixed hub graph). The generated DAG is the same as that in Example 1, but the conditional distribution of each node given its parents may be different. Particularly, we first generate distribution labels Z_j independently from $\text{Bern}(0.5)$ for $j = 1 \ldots, p$. Then the hub node X_1 in $\mathcal{A}_0$ is generated from $Z_1\text{Pois}(\exp(\theta_1)) + (1-Z_1)\text{Bin}(4, \theta_1)$, and X_j in $\mathcal{A}_1$ is generated from $Z_j\text{Pois}(\exp(\theta_j+\theta_{j1}X_1)) + (1-Z_j)\text{Bin}(4, \theta_j+\theta_{j1}X_1)$ for $j = 2, \ldots, p$. For those X_j's from the Poisson distribution, the parameters θ_j and θ_{jk} are generated uniformly from $[1, 3]$ and $[0.1, 0.3]$, $[0.1, 0.2]$ and $[0.05, 0.2]$ for $p = 5, 20, 100$, respectively; and for those X_j's from the Binomial distribution, the parameters θ_j and θ_{jk} are both generated uniformly from $[0.1, 0.2]$ for $p = 5$ and 20, and $[0.05, 0.2]$ for $p = 100$. Note that the shrunk interval for large p is used to avoid large values of $\theta_j + \theta_{j1}X_1$, which leads to a contradiction with Condition 1.

In each example, the averaged performance metrics of all the competing methods over 50 independent replications as well as their standard errors are summarized in Tables 1–2. It is evident that TLDAG yields superior numerical performance and outperforms the other five competitors in almost all the scenarios. In Table 1 with the Poisson hub graph, TLDAG yields a small HM and the largest Precision and F1-score, and the Recall of TLDAG, ODS and MRS are all close to 1, but the other three methods have much smaller Recall. In Table 2 with the mixed hub graph, TLDAG yields the best performance in terms of Precision and F1-score, and comparable performance to the best performer in terms of HM and Recall.

4.2. Random Graphs

We now consider two commonly used models for the random graphs, including the Erdös and Rényi (ER) model (Erdös and Rényi 1960) and the Barabási-Albert (BA) model (Barabási and Albert 1999). It is interesting to note that the BA model generates scale-free graphs, which commonly appears in many science problems, such as the gene networks.

Example 3 (Mixed ER graph). The generated DAG model is depicted in Figure 3. We set the probability of connecting an edge as $P_E = 0.35$ for $p = 5$ and 20, and $P_E = 0.1$ for $p = 100$ and generate a random DAG. Then, we convert the generated random DAG into topological structure, obtain the distribution labels $Z_j, j = 1, \dots, p$ as the same treatment in Example 2, and generate the data for the root nodes X_j's from $Z_j\text{Pois}(\exp(\theta_j)) + (1 - Z_j)\text{Bin}(4, \theta_j)$, and the remaining nodes X_k's from $Z_k\text{Pois}(\exp(\theta_k + \sum_{l\in \text{pa}_k} \theta_{kl}X_l)) + (1 - Z_k)\text{Bin}(4, \theta_k +$

Table 1. The averaged performance metrics of various methods as well as their standard errors in parentheses in Example 1.

p	n	Methods	HM	Recall	Precision	F1-score
5	200	TLDAG	0.13(0.01)	1.00(0.00)	0.63(0.01)	0.76(0.01)
		ODS	0.17(0.01)	1.00(0.01)	0.56(0.01)	0.71(0.01)
		MRS	0.14(0.01)	0.99(0.01)	0.58(0.01)	0.73(0.01)
		DLiNGAM	0.28(0.02)	0.56(0.04)	0.39(0.03)	0.45(0.03)
		GES	0.43(0.02)	0.34(0.03)	0.19(0.02)	0.25(0.02)
		MMHC	0.25(0.02)	0.54(0.05)	0.41(0.04)	0.46(0.04)
	500	TLDAG	0.14(0.01)	1.00(0.00)	0.62(0.02)	0.76(0.01)
		ODS	0.17(0.01)	1.00(0.00)	0.56(0.01)	0.71(0.01)
		MRS	0.13(0.01)	1.00(0.01)	0.59(0.01)	0.73(0.01)
		DLiNGAM	0.25(0.02)	0.59(0.04)	0.43(0.03)	0.49(0.03)
		GES	0.46(0.02)	0.31(0.03)	0.17(0.02)	0.22(0.02)
		MMHC	0.28(0.02)	0.56(0.04)	0.40(0.03)	0.44(0.03)
20	200	TLDAG	0.05(0.00)	1.00(0.00)	0.56(0.03)	0.70(0.02)
		ODS	0.11(0.00)	0.99(0.01)	0.32(0.01)	0.48(0.01)
		MRS	0.06(0.00)	1.00(0.00)	0.41(0.01)	0.58(0.01)
		DLiNGAM	0.29(0.01)	0.61(0.05)	0.10(0.01)	0.17(0.01)
		GES	0.18(0.00)	0.32(0.02)	0.11(0.01)	0.16(0.01)
		MMHC	0.09(0.00)	0.12(0.01)	0.12(0.01)	0.12(0.01)
	500	TLDAG	0.05(0.00)	1.00(0.00)	0.55(0.03)	0.69(0.02)
		ODS	0.11(0.00)	1.00(0.00)	0.32(0.01)	0.48(0.01)
		MRS	0.06(0.00)	0.99(0.00)	0.43(0.01)	0.60(0.01)
		DLiNGAM	0.25(0.01)	0.63(0.05)	0.12(0.01)	0.20(0.02)
		GES	0.19(0.01)	0.42(0.03)	0.12(0.01)	0.18(0.02)
		MMHC	0.10(0.00)	0.18(0.01)	0.13(0.01)	0.15(0.01)
100	200	TLDAG	0.02(0.00)	0.99(0.00)	0.44(0.03)	0.59(0.02)
		ODS	0.05(0.00)	0.94(0.01)	0.16(0.00)	0.28(0.01)
		MRS	0.02(0.00)	0.94(0.01)	0.30(0.01)	0.46(0.01)
		DLiNGAM	0.28(0.00)	0.54(0.04)	0.02(0.00)	0.04(0.00)
		GES	0.05(0.00)	0.09(0.01)	0.02(0.00)	0.03(0.00)
		MMHC	0.02(0.00)	0.01(0.00)	0.01(0.00)	0.01(0.00)
	500	TLDAG	0.02(0.00)	1.00(0.00)	0.45(0.03)	0.60(0.02)
		ODS	0.05(0.00)	0.97(0.00)	0.17(0.00)	0.29(0.00)
		MRS	0.02(0.00)	0.98(0.00)	0.33(0.01)	0.49(0.01)
		DLiNGAM	0.26(0.00)	0.59(0.03)	0.02(0.00)	0.04(0.00)
		GES	0.07(0.00)	0.17(0.01)	0.03(0.00)	0.05(0.00)
		MMHC	0.02(0.00)	0.03(0.00)	0.03(0.00)	0.03(0.00)

Table 2. The averaged performance metrics of various methods as well as their standard errors in parentheses in Example 2.

p	n	Methods	HM	Recall	Precision	F1-score
5	200	TLDAG	0.23(0.02)	0.57(0.04)	0.46(0.03)	0.50(0.03)
		ODS	0.33(0.01)	0.36(0.04)	0.26(0.03)	0.29(0.03)
		DLiNGAM	0.37(0.02)	0.47(0.05)	0.27(0.03)	0.34(0.03)
		GES	0.31(0.01)	0.41(0.03)	0.30(0.02)	0.34(0.02)
		MMHC	0.22(0.02)	0.55(0.04)	0.45(0.03)	0.49(0.03)
	500	TLDAG	0.24(0.02)	0.58(0.04)	0.47(0.04)	0.50(0.04)
		ODS	0.33(0.01)	0.39(0.04)	0.26(0.03)	0.31(0.03)
		DLiNGAM	0.34(0.02)	0.55(0.05)	0.32(0.03)	0.40(0.03)
		GES	0.31(0.01)	0.45(0.03)	0.31(0.02)	0.37(0.03)
		MMHC	0.23(0.02)	0.60(0.04)	0.45(0.03)	0.51(0.03)
20	200	TLDAG	0.12(0.01)	0.27(0.03)	0.18(0.02)	0.21(0.02)
		ODS	0.12(0.01)	0.23(0.03)	0.10(0.01)	0.13(0.02)
		DLiNGAM	0.24(0.01)	0.27(0.03)	0.06(0.01)	0.10(0.01)
		GES	0.11(0.00)	0.29(0.01)	0.17(0.01)	0.21(0.01)
		MMHC	0.09(0.00)	0.12(0.01)	0.11(0.01)	0.11(0.01)
	500	TLDAG	0.12(0.01)	0.31(0.03)	0.20(0.03)	0.23(0.02)
		ODS	0.14(0.01)	0.27(0.03)	0.11(0.01)	0.14(0.02)
		DLiNGAM	0.25(0.01)	0.33(0.03)	0.08(0.01)	0.13(0.01)
		GES	0.10(0.00)	0.37(0.02)	0.20(0.01)	0.26(0.01)
		MMHC	0.09(0.00)	0.22(0.01)	0.17(0.01)	0.19(0.01)
100	200	TLDAG	0.05(0.00)	0.22(0.02)	0.09(0.02)	0.12(0.02)
		ODS	0.06(0.00)	0.21(0.02)	0.04(0.01)	0.07(0.01)
		DLiNGAM	0.18(0.01)	0.40(0.02)	0.02(0.00)	0.04(0.00)
		GES	0.03(0.00)	0.12(0.01)	0.05(0.00)	0.07(0.01)
		MMHC	0.02(0.00)	0.02(0.00)	0.01(0.00)	0.02(0.00)
	500	TLDAG	0.05(0.00)	0.29(0.03)	0.10(0.02)	0.14(0.02)
		ODS	0.07(0.00)	0.26(0.02)	0.05(0.01)	0.07(0.01)
		DLiNGAM	0.15(0.01)	0.31(0.02)	0.02(0.00)	0.04(0.00)
		GES	0.04(0.00)	0.19(0.02)	0.06(0.01)	0.10(0.01)
		MMHC	0.02(0.00)	0.03(0.00)	0.02(0.00)	0.03(0.00)

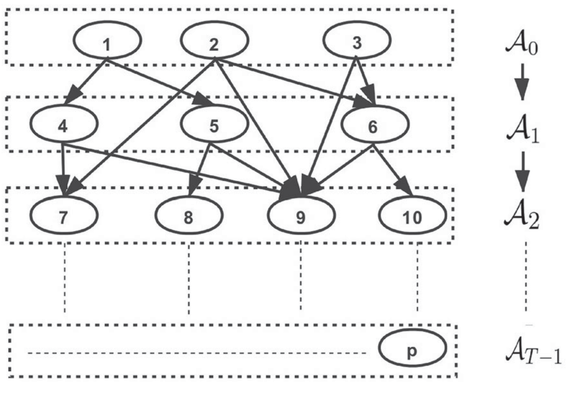

Figure 3. The random graph for Examples 3 and 4.

$\sum_{l \in \mathrm{pa}_k} \theta_{kl} X_l)$. Precisely, the parameter θ_k for the Poisson distribution is generated uniformly from $[1, 3]$, and θ_{kl} are generated uniformly from $[0.01, 0.05]$, $[0.005, 0.015]$, and $[0.001, 0.01]$ for $p = 5, 20$, and 100, respectively. The parameter θ_k and θ_{kl} for the Binomial distribution are all generated uniformly from $[0.01, 0.05]$, $[0.005, 0.015]$, and $[0.005, 0.01]$ for $p = 5, 20$, and 100, respectively.

Example 4 (Mixed BA graph). The generated DAG is a BA graph with the number of edges to be added for each node as 2. The generated data mechanism is the same as that in Example 3 except that the parameter θ_k for the Poisson distribution is generated uniformly from $[1, 3]$, and θ_{kl} are generated uniformly from $[0.01, 0.03]$, $[0.005, 0.02]$, and $[0.001, 0.01]$ for $p = 5, 20$, and 100, respectively. The parameter θ_k and θ_{kl} for the Binomial distribution are all generated uniformly from $[0.01, 0.05]$, $[0.005, 0.02]$, and $[0.001, 0.01]$ for $p = 5, 20$, and 100, respectively.

In each example, the averaged performance metrics of all the competing methods over 50 independent replications as well as their standard errors are summarized in Tables 3 and 4. Note that the averaged number of topological layers considered in Examples 3 and 4 for the case $p = 100$ are as large as 18 and 9, respectively.

From Tables 3 and 4, it is clear that TLDAG still performs well on the random graphs. Precisely, in the both two examples, TLDAG is the best performer in terms of HM and Precision, and yields comparable performance to the other competitors in terms of F1-score in almost all the scenarios. Note that TLDAG achieves the highest Precision in all scenarios, largely due to the fact that TLDAG can obtain high layer-recovery accuracy, leading to better DAG estimation accuracy. It is worthy pointing out that DLiNGAM achieves the highest Recall in some cases, due to the fact that it tends to produce a very dense graph with many false edges, leading to small Precision. Note that the MMHC method does not return any result after running for more than 24 hr for the cases with $p = 100$ in Example 4, and thus is omitted in Table 4 correspondingly.

4.3. Computational Comparison

We now turn to investigate the computational efficiency of TLDAG and compare it with ODS and MRS. Specifically, their running time is measured with varying n and p under two cases that: (1) $n \in \{100, 500, 1000, 1500, 2000, 2500\}$ with $p = 50$ in the hub Poisson and mixed BA graphs; (2) $p \in \{10, 20, 40, 60, 80, 100\}$ with $n = 500$ in the hub and Poisson BA graphs. The data-generating scheme is the same as that in Sections 4.1 and 4.2 and the running time is reported in Figure 4. We also evaluate the running time of these algorithms in a Poisson graph with different cardinality of each layer that $a_t \in \{5, 10, 15, 20, 25\}$ under the case $(n, p) = (500, 50)$, and the number of parent of each node to be one. Figure 5 illustrates the obtained results.

It is evident that TLDAG is much more efficient than other methods in terms of computational cost, where all the tuning

Table 3. The averaged performance metrics of various methods as well as their standard errors in parentheses in Example 3.

p	n	Methods	HM	Recall	Precision	F1-score
5	200	TLDAG	0.27(0.02)	0.34(0.04)	0.34(0.05)	0.31(0.04)
		ODS	0.32(0.01)	0.25(0.03)	0.22(0.04)	0.22(0.03)
		DLiNGAM	0.41(0.02)	0.39(0.05)	0.19(0.02)	0.24(0.03)
		GES	0.31(0.01)	0.37(0.04)	0.25(0.03)	0.28(0.03)
		MMHC	0.29(0.01)	0.28(0.03)	0.26(0.03)	0.25(0.03)
	500	TLDAG	0.24(0.02)	0.39(0.04)	0.41(0.05)	0.37(0.04)
		ODS	0.32(0.01)	0.29(0.03)	0.24(0.03)	0.25(0.03)
		DLiNGAM	0.39(0.02)	0.40(0.05)	0.20(0.02)	0.25(0.03)
		GES	0.31(0.01)	0.38(0.05)	0.25(0.03)	0.28(0.03)
		MMHC	0.29(0.01)	0.33(0.04)	0.26(0.03)	0.27(0.03)
20	200	TLDAG	0.20(0.01)	0.12(0.01)	0.37(0.03)	0.16(0.01)
		ODS	0.23(0.00)	0.12(0.01)	0.22(0.01)	0.15(0.01)
		DLiNGAM	0.31(0.01)	0.27(0.02)	0.20(0.01)	0.22(0.01)
		GES	0.23(0.00)	0.17(0.01)	0.26(0.01)	0.20(0.01)
		MMHC	0.21(0.00)	0.07(0.01)	0.20(0.01)	0.11(0.01)
	500	TLDAG	0.21(0.01)	0.14(0.01)	0.36(0.03)	0.17(0.02)
		ODS	0.23(0.01)	0.13(0.01)	0.22(0.02)	0.15(0.01)
		DLiNGAM	0.31(0.01)	0.27(0.02)	0.21(0.01)	0.22(0.01)
		GES	0.23(0.00)	0.22(0.02)	0.27(0.01)	0.24(0.01)
		MMHC	0.21(0.00)	0.11(0.01)	0.24(0.01)	0.15(0.01)
100	200	TLDAG	0.07(0.00)	0.03(0.00)	0.15(0.02)	0.04(0.00)
		ODS	0.10(0.00)	0.05(0.01)	0.05(0.00)	0.05(0.00)
		DLiNGAM	0.24(0.01)	0.24(0.01)	0.06(0.00)	0.08(0.00)
		GES	0.08(0.00)	0.09(0.00)	0.11(0.00)	0.10(0.00)
		MMHC	0.06(0.00)	0.02(0.00)	0.08(0.00)	0.03(0.00)
	500	TLDAG	0.08(0.00)	0.06(0.01)	0.10(0.01)	0.06(0.00)
		ODS	0.12(0.00)	0.08(0.01)	0.05(0.00)	0.06(0.00)
		DLiNGAM	0.18(0.00)	0.18(0.01)	0.06(0.00)	0.09(0.00)
		GES	0.09(0.00)	0.14(0.00)	0.13(0.00)	0.14(0.00)
		MMHC	0.07(0.00)	0.04(0.00)	0.10(0.00)	0.05(0.00)

Table 4. The averaged performance metrics of various methods as well as their standard errors in parentheses in Example 4.

p	n	Methods	HM	Recall	Precision	F1-score
5	200	TLDAG	0.28(0.02)	0.40(0.03)	0.73(0.05)	0.49(0.04)
		ODS	0.35(0.02)	0.36(0.03)	0.49(0.04)	0.41(0.03)
		DLiNGAM	0.47(0.02)	0.38(0.03)	0.35(0.03)	0.36(0.03)
		GES	0.39(0.01)	0.33(0.03)	0.44(0.05)	0.37(0.04)
		MMHC	0.42(0.01)	0.23(0.02)	0.35(0.05)	0.27(0.03)
	500	TLDAG	0.28(0.01)	0.47(0.03)	0.70(0.04)	0.54(0.03)
		ODS	0.37(0.02)	0.35(0.03)	0.47(0.03)	0.39(0.03)
		DLiNGAM	0.45(0.02)	0.39(0.04)	0.37(0.04)	0.37(0.03)
		GES	0.40(0.02)	0.36(0.04)	0.42(0.05)	0.38(0.04)
		MMHC	0.43(0.01)	0.25(0.03)	0.34(0.04)	0.29(0.03)
20	200	TLDAG	0.12(0.01)	0.15(0.01)	0.49(0.05)	0.21(0.02)
		ODS	0.18(0.00)	0.08(0.02)	0.09(0.02)	0.08(0.02)
		DLiNGAM	0.25(0.01)	0.27(0.02)	0.12(0.01)	0.17(0.01)
		GES	0.17(0.00)	0.17(0.01)	0.16(0.02)	0.17(0.01)
		MMHC	0.15(0.00)	0.08(0.01)	0.12(0.01)	0.10(0.01)
	500	TLDAG	0.13(0.01)	0.18(0.03)	0.48(0.05)	0.24(0.02)
		ODS	0.19(0.01)	0.08(0.02)	0.08(0.02)	0.08(0.02)
		DLiNGAM	0.25(0.01)	0.28(0.03)	0.13(0.01)	0.18(0.01)
		GES	0.17(0.00)	0.21(0.02)	0.18(0.02)	0.20(0.02)
		MMHC	0.15(0.00)	0.10(0.01)	0.14(0.02)	0.12(0.01)
100	200	TLDAG	0.03(0.00)	0.04(0.01)	0.33(0.05)	0.06(0.01)
		ODS	0.10(0.01)	0.05(0.01)	0.01(0.01)	0.02(0.01)
		DLiNGAM	0.13(0.00)	0.19(0.01)	0.03(0.00)	0.05(0.00)
		GES	0.05(0.00)	0.08(0.00)	0.05(0.00)	0.06(0.00)
		MMHC	–	–	–	–
	500	TLDAG	0.04(0.01)	0.06(0.01)	0.29(0.04)	0.08(0.01)
		ODS	0.13(0.01)	0.07(0.01)	0.01(0.00)	0.02(0.00)
		DLiNGAM	0.12(0.00)	0.17(0.02)	0.03(0.01)	0.05(0.01)
		GES	0.05(0.00)	0.11(0.00)	0.06(0.01)	0.08(0.00)
		MMHC	–	–	–	–

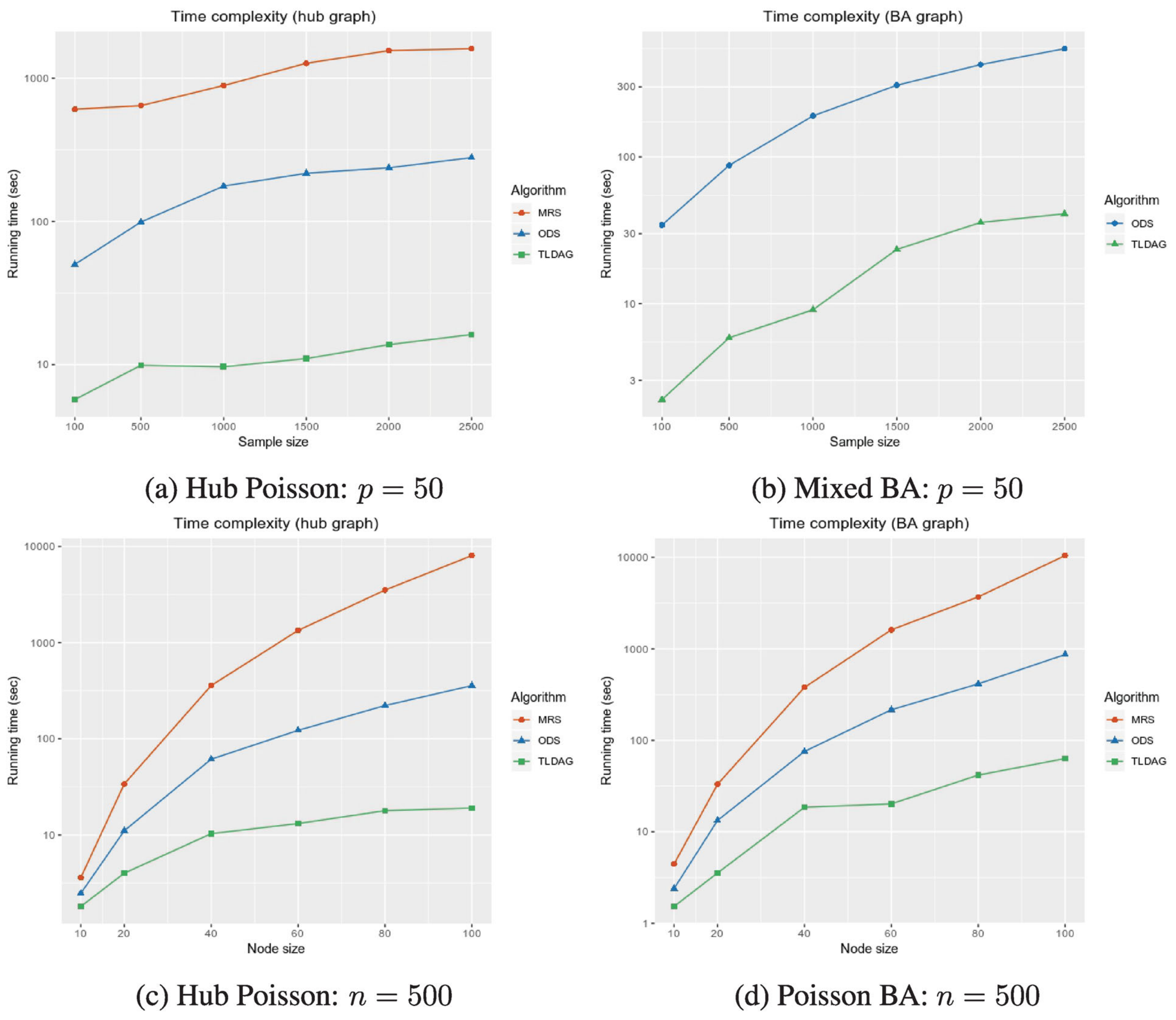

Figure 4. Comparisons of the TLDAG algorithm with the ODS and MRS algorithms in terms of the running time with respect to varying (a) the sample size in a hub Poisson graph, (b) the sample size in a mixed BA graph, (c) the node size in a hub Poisson graph, (d) the node size in a Poisson BA graph.

procedures are taken into consideration. Specifically, the computational cost of TLDAG is roughly linearly proportional to the sample size and the node size. It is also interesting to point out that the computational cost of these algorithms is not sensitive to the cardinality of each layer in Figure 5. This is reasonable as the nodes in each layer can be simultaneously identified by TLDAG, whereas both ODS and MRS are only designed to estimate the causal ordering. The computational efficiencies reported in Figures 4 and 5 also support the computational complexity analysis in Section 3.2.

5. Real Applications

We now apply the TLDAG algorithm to analyze two real examples, including an NBA player statistics data and a cosmetic sales data. The NBA player statistics data is publicly available in the R package "SportsAnalytics," and the cosmetic sales data is collected by Alibaba, one of the largest online stores in China.

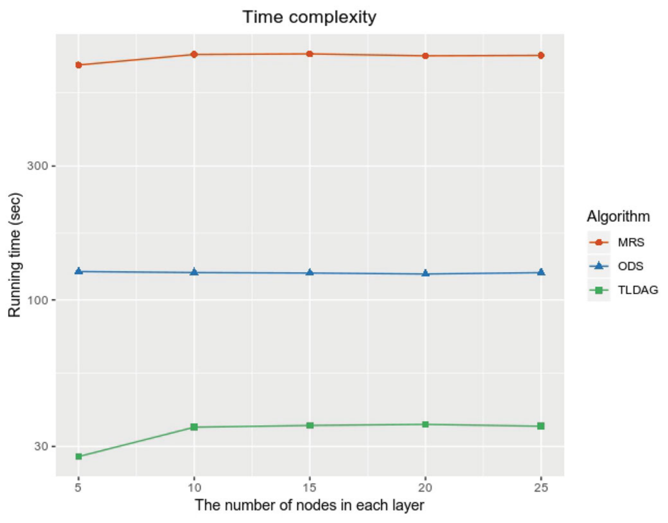

Figure 5. Comparisons of the TLDAG algorithm with the ODS and MRS algorithms in terms of the running time with the number of nodes in each layer.

5.1. NBA Player Data

The NBA player statistics data consists of a number of statistics for 441 NBA players in the season 2009/2010. For illustration, we focus on 18 informative statistics, including TotalMinutesPlayed, FieldGoalsMade, FieldGoalsAttempted, ThreesMade, ThreesAttempted, FreeThrowsMade, FreeThrowsAttempted, OffensiveRebounds, TotalRebounds, Assists, Steals, Turnovers, Blocks, PersonalFouls, Disqualifications, TotalPoints, Technicals, and GamesStarted.

Following the same treatment as in Park and Raskutti (2018), we assume the conditional distribution of each node given its parents follows a Poisson distribution. We then apply TLDAG to estimate the directed structures among the 18 statistics, as shown in Figure 6.

The estimated DAG in Figure 6 has four topological layers and twenty directed edges. Compared with the estimated DAG by ODS, it contains six more directed edges and reverses six directed edges, and the difference is summarized in Figure 7.

It is evident that TLDAG produces a much more reasonable DAG compared with ODS. The added edges appear reasonable and agree with common sense. The more GamesStarted and the less Turnovers a player has, the more FieldGoalsMade and TotalRebounds he may obtain by controlling and dribbling more balls; if a player has competitive ability in Steals and Assists and makes more three point shots, he is more likely to be a key player in the team and thus plays more minutes in the game. For the reversed edges, it is reasonable that a larger number of FieldGoalMade implies more total points, but the reverse is not necessarily true; if a player is in center or Power Forward position, he is likely to have more PersonalFouls, also more OffensiveRebounds, and thus more TotalRebouds. Note that the position variable is seen as a latent variable and excluded in this graph. Furthermore, strong ability of taking more TotalRebounds for a player leads to his more minutes to play, and more FieldGoalsAttempted and less PersonalFouls show the player's offensive and defensive abilities, which results in more minutes he plays in the game.

5.2. Alibaba Cosmetic Sales Data

The cosmetic sales dataset consists of 35,102 samples and six features of liquid essences of some anonymous brands, including the number of orders in one month (NO), the level of brand (LB), the star level of the seller (SLS), the place of origin (PO), the effect (EF) and the ingredient (IND). Note that NO is a continuous variable and the other five variables are discrete. Specifically, LB and SLS take values in $\{0, 1, 2, 3, 4, 5\}$ and $\{0, 1, 2, 3\}$, where a larger value indicates higher reputation for the brand and seller. PO takes value in $\{0, 1, 2, \ldots, 21\}$ representing 22 different countries of origin, EF takes value in $\{0, 1, 2, \ldots, 31\}$ for effects including moisurization, skin whitening, despeckle and so on, and IND takes value in $\{0, 1, 2, \ldots, 122\}$ for different ingredients such as water, polyois, essence, solubilizer and so on.

As suggested by the descriptive statistics, it is reasonable to assume that the conditional distribution of NO given its parents follows an exponential distribution, and the conditional distributions of the other five discrete variables given their parents are Binomial. We then apply the TLDAG algorithm to the dataset and the estimated DAG is shown in the left panel of Figure 8, which has four topological layers and 10 directed edges.

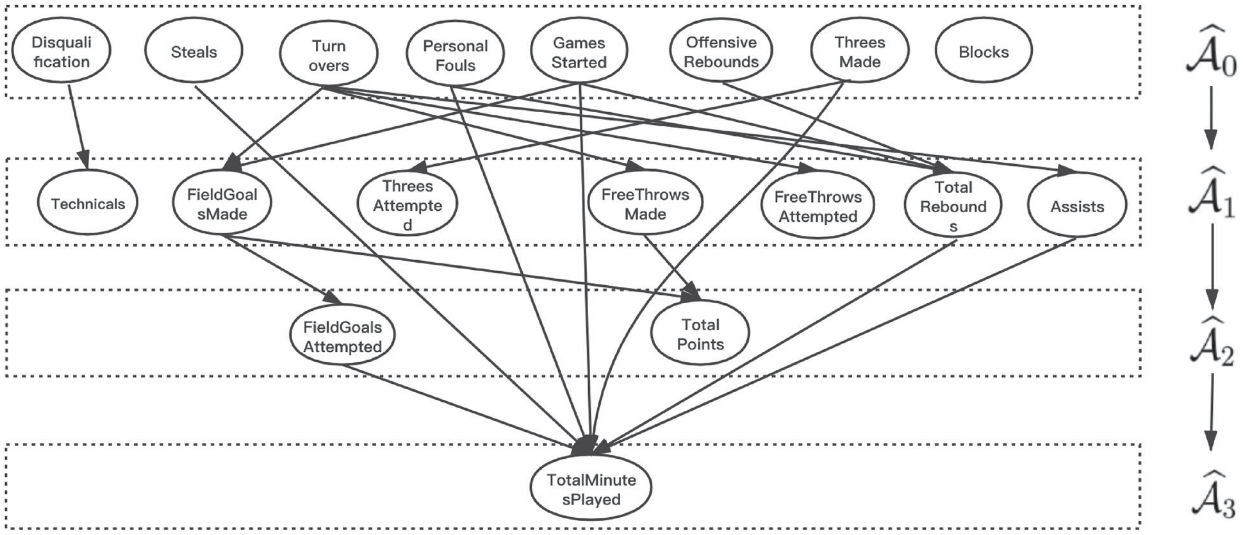

Figure 6. The estimated DAG among 18 statistics in the NBA player data.

Added Edges	GamesStarted $\rightarrow$ FieldGoalsMade / TotalRebounds, Steals / Assists / ThreesMade $\rightarrow$ TotalMinutesPlayed, Turnovers $\rightarrow$ FieldGoalsMade.
Reversed Edges	FieldGoalsMade $\rightarrow$ TotalPoints, PersonalFouls/OffensiveRebounds $\rightarrow$ TotalRebounds, FieldGoalsAttempted/PersonalFouls/TotalRebounds $\rightarrow$ TotalMinutesPlayed.

Figure 7. The difference between the estimated DAGs by TLDAG and ODS.

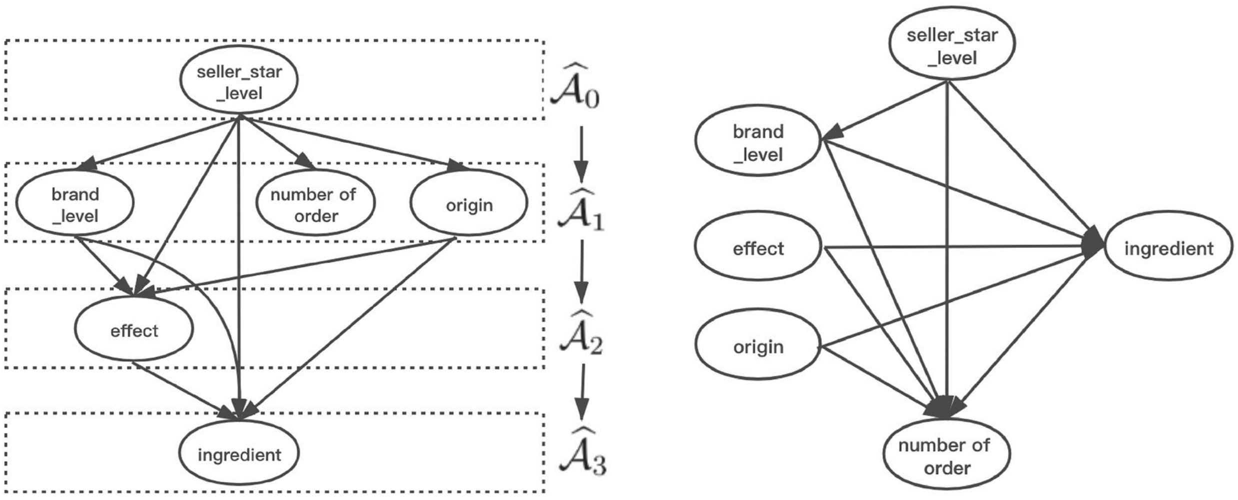

Figure 8. The estimated DAGs in the cosmetic sales data by TLDAG (left) and ODS (right).

In Figure 8 , some of estimated directed edges are highly interpretable. For example, a higher star level of the seller indicates a better reputation and higher customer loyalty, which leads to a larger number of orders; different origins have their unique materials and ingredients for the specific effect; the higher the brand level is, the more various and advanced effects the liquid essence has. The last two directed edges, however, are missed in the estimated DAG by ODS, shown in the right panel of Figure 8. It is also interesting to point out that the imposed conditional distribution assumptions may not be satisfied by the cosmetic sales data, and thus some of estimated edges are not easy to interpret, such as the seller level $\rightarrow$ the brand level/origin.

6. Summary

In this article, we propose a computationally efficient learning algorithm for a large class of non-Gaussian DAGs, denoted as QVF-DAGs. The proposed algorithm is based on a novel concept of topological layers, and consists of two steps of learnings. It first reconstructs the topological layers in a hierarchical fashion, and then reconstructs the directed edges between nodes in different layers. The computational complexity of the proposed algorithm is much less than the existing learning algorithms in literature. The computational efficiency and the estimation accuracy of the proposed method are also supported by a number of simulated examples and two real applications.

Supplementary Materials

The supplementary files include computational details, all the theoretical results, and the corresponding proofs.

Acknowledgments

The authors thank the editor, the associate editor, and the three anonymous referees for their constructive suggestions, which significantly improve this article.

Funding

Xin He's research is supported in part by NSFC-11901375 and Shanghai Pujiang Program 2019PJC051, Wei Zhong's research is supported in part by NSFC-71988101 and NSFC-11922117, and Junhui Wang's research is supported in part by HK RGC Grants GRF-11300919, GRF-11304520 and GRF-11206821.

References

Barabási, A., and Albert, R. (1999), "Emergence of Scaling in Random Networks," *Science*, 286, 509–512. [6]

Brown, L. D., Cai, T. T., and Zhou, H. H. (2010), "Nonparametric Regression in Exponential Families," *The Annals of Statistics*, 38, 2005–2046. [2]

Bühlmann, P., Peters, J., and Ernest, J. (2014), "CAM: Causal Additive Models, High-Dimensional Order Search and Penalized Regression," *The Annals of Statistics*, 42, 2526–2556. [1]

Chen, W. Y., Drton, M., and Wang, Y. S. (2019), "On Causal Discovery with an Equal-Variance Assumption," *Biometrika*, 106, 973–980. [1]

Chickering, D. W. (2003), "Optimal Structure Identification with Greedy Search," *The Journal of Machine Learning Research*, 3, 507–554. [1,5]

Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C. (2009), *Introduction to Algorithms*, Cambridge, MA: MIT Press. [2]

Erdös, P., and Rényi, A. (1960), "On the Evolution of Random Graphs," *Publications of the Mathematical Institute of the Hungarian Academy of Sciences*, 5, 17–60. [6]

Friedman, J., Hastie, T., and Tibshirani, R. (2010), "Regularization Paths for Generalized Linear Models via Coordinate Descent," *Journal of Statistical Software*, 33, 1–22. [5]

Heckerman, D., Geiger, D., and Chickering, D. M. (1995), "Learning Bayesian Networks: The Combination of Knowledge and Statistical Data," *Machine Learning*, 20, 197–243. [3]

Kalisch, M., and Bühlmann, P. (2007), "Estimating High-Dimensional Directed Acyclic Graphs with the PC-Algorithm," *The Journal of Machine Learning Research*, 8, 613–636. [1]

Kalisch, M., Mächler, M., Colombo, D., Maathuis, M. H., and Bühlmann, P. (2012), "Causal Inference Using Graphical Models with the R package pcalg," *Journal of Statistical Software*, 47, 1–26. [5]

Koller, D., and Friedman, N. (2009), *Probabilistic Graphical Models: Principles and Techniques*, Cambridge, MA: MIT Press. [1]

Meinshausen, N., and Bühlmann, P. (2006), "High-Dimensional Graphs and Cariable Selection with the Lasso," *The Annals of Statistics*, 34, 1436–1462. [4]

Morris, C. N. (1982), "Natural Exponential Families with Quadratic Variance Functions," *The Annals of Statistics*, 10, 65–80. [2]

Nandy, P., Hauser, A., and Maathuis, M. H. (2018), "High-Dimensional Consistency in Score-Based and Hybrid Structure Learning," *The Annals of Statistics*, 46, 3151–3183. [1]

Park, G., and Park, S. (2019), "High-Dimensional Poisson Structural Equation Model Learning via ℓ_1-Regularized Regression," *The Journal of Machine Learning Research*, 18, 1–41. [1,3,4,5]

Park, G., and Raskutti, G. (2018), "Learning Quadratic Variance Function DAG Models via Overdispersion Scoring," *The Journal of Machine Learning Research*, 18, 1–44. [1,2,3,4,9]

Peters, J., and Bühlmann, P. (2014), "Identifiability of Gaussian Structural Equation Models with Equal Error Variances," *Biometrika*, 101, 219–228. [1]

Peters, J., Mooij, J. M., Janzing, D., and Schölkopf, B. (2014), "Causal Discovery with Continuous Additive Noise Models," *The Journal of Machine Learning Research*, 15, 2009–2053. [1]

Sachs, K., Perez, O., Peer, D., Lauffenburger, D. A., and Nolan, G. P. (2005), "Causal Protein-Signaling Networks Derived from Multiparameter Single-Cell Data," *Science*, 308, 523–529. [1]

Sanford, A. D., and Moosa, I. A. (2012), "A Bayesian Network Structure for Operational Risk Modelling in Structured Finance Operations," *Journal of the Operational Research Society*, 63, 431–444. [1]

Scutari, M. (2010), "Learning Bayesian Networks with the bnlearn R package," *Journal of Statistical Software*, 35, 1–22. [5]

Shimizu, S., Hoyer, P. O., Hyvärinen, A., and Kerminen, A. J. (2006), "A Linear Non-Gaussian Acyclic Model for Causal Discovery," *The Journal of Machine Learning Research*, 7, 2003–2030. [1]

Shimizu, S., Inazumi, T., Sogawa, Y., Hyvärinen, A., Kawahara, Y., Washio, T., Hoyer, P. O., and Bollen, K. (2011), "DirectLiNGAM: A Direct Method for Learning a Linear Non-Gaussian Structural Equation Model," *The Journal of Machine Learning Research*, 12, 1225–1248. [1,5]

Spirtes, P., Glymour, C. N., and Scheines, R. (2000), *Causation, Prediction, and Search*, Cambridge, MA: MIT Press. [1]

Sun, W. W., Wang, J. H., and Fang, Y. X. (2013), "Consistent Selection of Tuning Parameters via Variable Selection Stability," *The Journal of Machine Learning Research*, 14, 3419–3440. [5]

Tsamardinos, I., Brown, L. E., and Aliferis, C. F. (2006), "The Max-Min Hill-Climbing Bayesian Network Structure Learning Algorithm," *Machine Learning*, 65, 31–78. [1,5]

Wang, Y. S., and Drton, M. (2020), "High-Dimensional Causal Discovery Under Non-Gaussianity," *Biometrika*, 107, 41–59. [1]

Yang, E., Ravikumar, P., Allen, G. I., and Liu, Z. D. (2015), "Graphical Models via Univariate Exponential Family Distributions," *The Journal of Machine Learning Research*, 16, 3813–3847. [4]

Yuan, Y. P., Shen, X. T., Pan, W., and Wang, Z. Z. (2019), "Constrained Likelihood for Reconstructing a Directed Acyclic Gaussian Graph," *Biometrika*, 106, 109–125. [1,5]

Zheng, X., Aragam, B., Ravikumar, P. K., and Xing, E. P. (2018), "DAGs with NO TEARS: Continuous Optimization for Structure Learning," in *Advances in Neural Information Processing Systems (NIPS)*, pp. 9472–9483. [1]

Zhu, S. Y., Ng, I., and Chen, Z. T. (2020), "Causal Discovery with Reinforcement Learning," in *International Conference on Learning Representations (ICLR)*. [1]